



Ecksdee?

Futuristic Racer made with CS and CEL

Libre software licensed under the GNU GPL.



Content Table

- * History of Ecksdee
- * Hovering
- * Crystal Core symbiosis
- * Quests for game logic
- * Ship behaviour
- * Application
- * Packaging
- * XML
- * Art
- * The Future



History

Other `wipeout' projects I worked on before.

Originally making the code for cc.

Crappy demo.

Vincent repackaging demo as an app.

Asking whether it could be a game.

* So it became a game *



Hovering

#1

Soft body underneath the ship

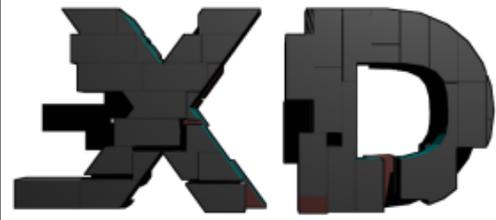
#2

Reactionary upthrust + angular correction

#3

4 Reactionary upthrust's

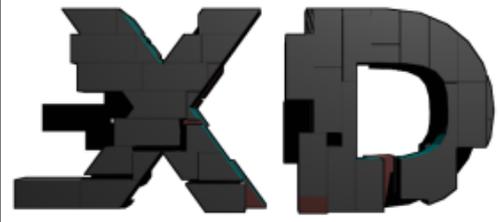
In the end we chose #2



Hover/Craft CEL

Jorrit wanted the hover code as CEL propclass
Better for us since it's a cleaner interface
Reusable for others (CrystalCore)
Controller interface

```
pccraft->SetTopSpeed (X)  
pccraft->SetTurningSpeed (Y)  
...
```



CC Symbiosis

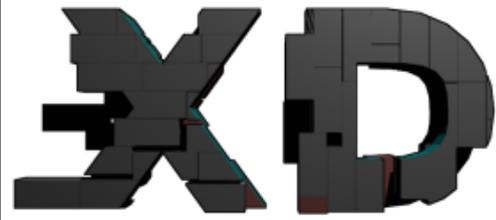
Reusing CC subsystems

- * Screens (start menu & game HUD)
- * Behaviours
- * Weapon
- * Others



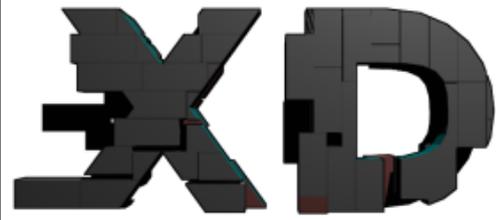
Quests usage

- * CEL Quest Manager for game logic
- * All XML but C++ specific inventory reward
- * Checkpoint quest: 2 states
- * Race quest: monitoring & storing properties
- * Others: updating the HUD



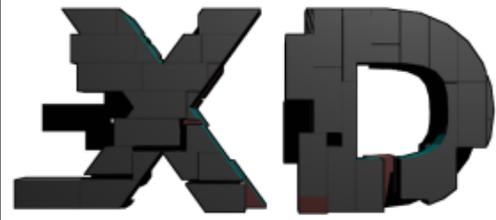
Ship behaviour

- * Handles messages from HUD, quests, gm, ...
 - Player input
 - Damage
 - Item pickup (pctimer)
 - Teleportation
 - TickEveryFrame() for sound pitching



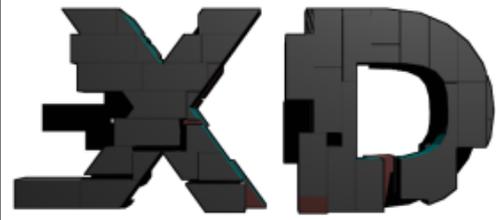
Application

- * Level loading (HUD & map)
- * Ship-specific config file loading
- * Setting of property classes
 - camera modes (pcdefaultcamera)
 - ship mesh object (pcmesh)
 - input keys (pccommandinput)
 - physics (pcmechanicsobject)
 - pchover & pccraft
 - config value stored in pcproperties



Application (2)

- * CS & CEL initialization
- * Creating entity list from zip files
- * FinishFrame(): csPen for thrust&energy bars
- * Screen system



Packaging

- * Ready-to-run executables are important
- * Requirements:
 - CS & CEL dependencies (zlib, ODE, ...)
 - CS & CEL core libraries
 - CS & CEL plugins (.dll/.so files)
 - Configuration files (CS/data/config/)
 - Shader files (CS/data/shader/)
 - Virtual File System : vfs.cfg



Packaging (2)

- * Ecksdee uses static plugin linking
 - > jam staticplugins
 - > specific Jamfile section to use them
- * Windows: provide CS & CEL dependencies like libz-cs.dll and libode-cs.dll, Cg ...
- * GNU/Linux: moving the dependencies' shared lib out of search dir so that CS uses static version (useful for ODE)



Xml

hero

```
<finished_laps>  
<required_laps>  
<countdown>  
<totalracetime>  
<totalenergy>  
<energy>  
<totalturbo>  
<turbo>  
...
```

Not necessarily initialized here



Xml (2)

```
<event name="pctimer_wakeup">
  <!-- Show ship's countdown every second if race is started -->
  <if eval="property(pctag('Checkpoint1',pcquest,racequest),propid(state))=='started'">
    <var entity="hero" name="countdown" value="?hero.countdown-1000" />
    <expr eval="hud_racestate_display::setcountdown(countdown=?hero.countdown)" />
    <if eval="?racestarted && !?raceend">
      <var entity="hero" name="totalracetime" value="?hero.totalracetime + 1" />
    </if>
    <if eval="?hero.countdown<=6000">
      <expr eval="hud_racestate_display::setstatus(status=?hero.countdown/1000)" />
    </if>
    <if eval="?hero.countdown<=1">
      <var name="racetimeout" value="true" />
      <expr eval="hero::xd_damage(amount=5000,source='ent_scene',sector='',position=[0,0,0],type='')" />
    </if>
    <expr eval="hud_racestate_display::updatehud_racestate()" />
  </if>
</event>
```



Xml (3)

Xml Script

```
<event name="pcbillboard_select">  
  <if eval="!?menu_paused">  
    </if>  
  </event>
```

C++

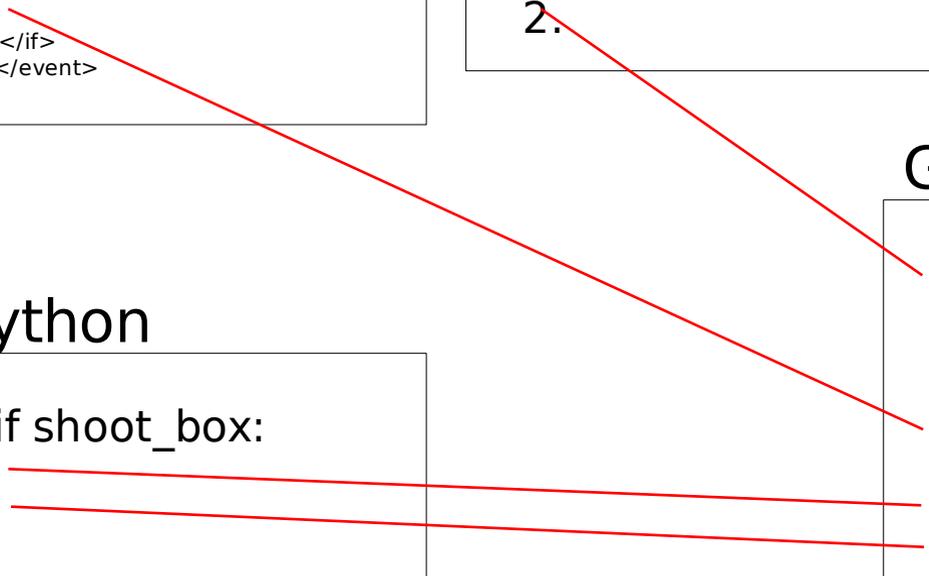
1. Application start ()
- 2.

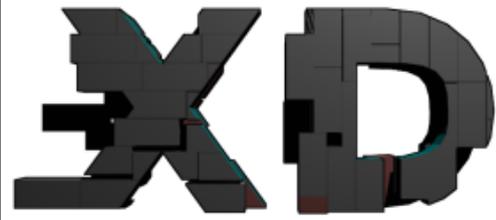
Python

```
if shoot_box:
```

Game Manager

```
StartRace ()  
ExitRace ()  
ExitApp ()  
Pause ()  
Unpause ()  
DoExplosion ()  
DoAreaDamage ()  
...
```





Art

"Minimalism describes movements in various forms of art and design, especially visual art and music, where the work is stripped down to its most fundamental features and core self expression."

~ Wikipedia article on minimalism

Interface is simplified wherever possible.
Less is more.

XD

Art (2)



XD

Art (3)





Art (4)





Art (5)

Hard to lay down restrictions with OSS

Art and Content for Ecksdee at the moment
is our primary goal



The Future

Better weapon system

Configuration setting from main menu

Placing checkpoints and bonuses in
blender2crystal

Hall of fame, track records and difficulty
modes

Ship upgrades



The Future

Networking

Artificial Intelligence